Tracking, Teil 1: Einführung

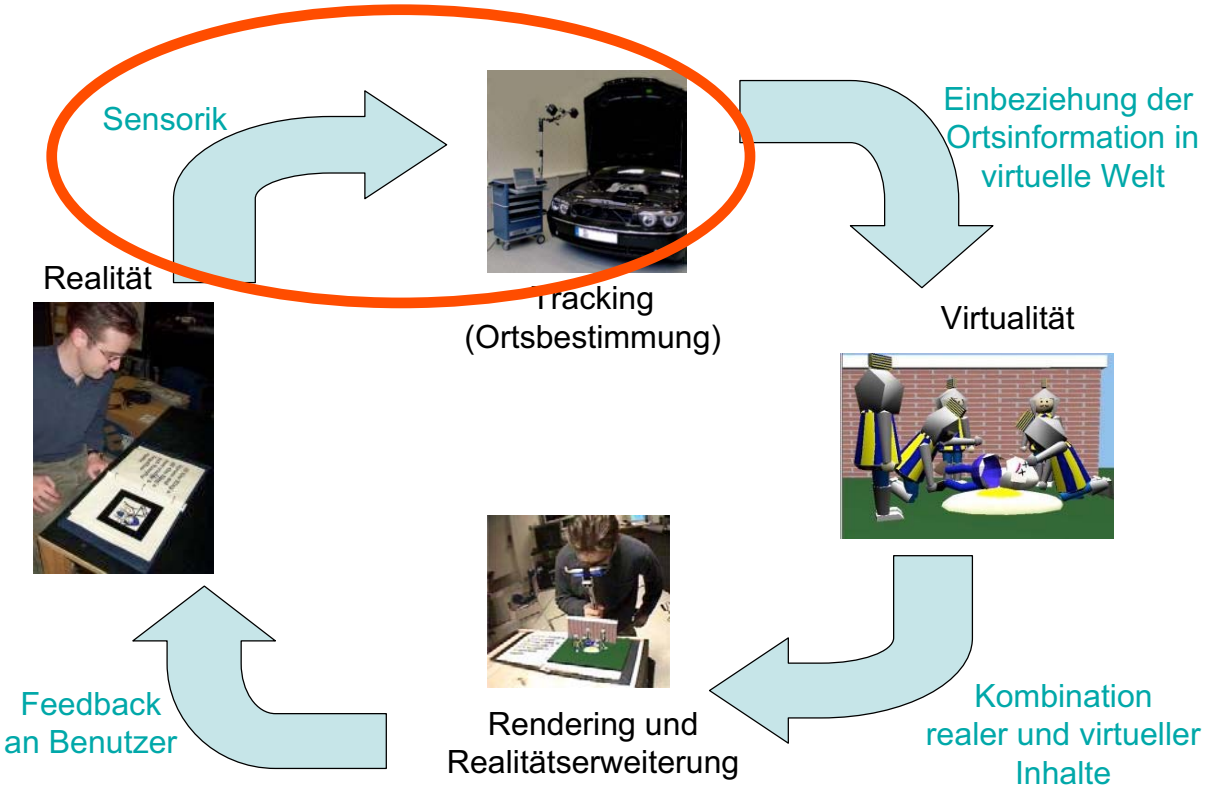
Vorlesung „Augmented Reality”

Prof. Dr. Andreas Butz

WS 2006/2007

Folien heute überw. von Dr. Martin Wagner

Ein Generisches AR-System



Überblick

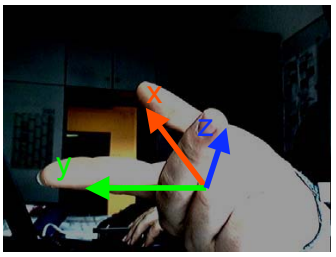
- Was ist Tracking?
- Koordinaten im 3D-Raum
- Transformationen im 2D-Raum
- Transformationen im 3D-Raum
- Darstellung von Rotationen im 3D-Raum
- Kombination von Transformationen
- Homogene Koordinaten
- Räumliche Beziehungen
- Mögliche Fehlerquellen beim Tracking

Was ist Tracking?

- *Korrekte Registrierung* zwischen realen und virtuellen Objekten zwingend für AR
- *Kalibrierung* ist (offline-) Bestimmung aller unveränderlichen Parameter, z.B.
  - Position fester Sensoren
  - Feste Eigenschaften von Kameras (z.B. Brennweite)
- *Tracking* ist (online-) Bestimmung zeitlich variabler Parameter, meist Position von
  - Benutzern
  - Realen Objekten

Koordinaten im 3D-Raum

- Kartesische Koordinaten
  - *Orthogonales* System
  - Koordinatenlinien sind Geraden mit konstantem Abstand
  - Achtung: *Einheit* beachten
- 2 Möglichkeiten (Obacht!):
  - *Links*händiges System
  - *Rechts*händiges System
    → in Zukunft verwendet



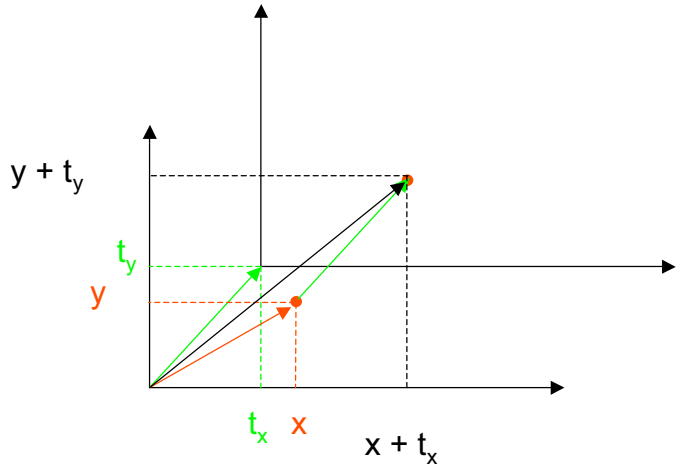
Affine Transformationen

- Bewahrung von Parallelität
- Kein Bewahrung von Winkeln und Längen
- Arten:
  - Translation
  - Rotation
  - Skalierung
  - Scheren

Transformationen im 2D-Raum (1)

- Punktdarstellung: $p = \begin{pmatrix} x \\ y \end{pmatrix}$

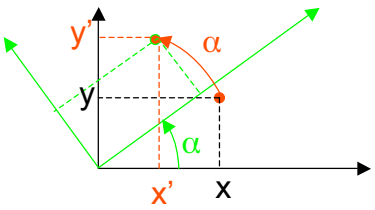
- Translation: $p' = p + t = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \end{pmatrix}$



Transformationen im 2D-Raum (2)

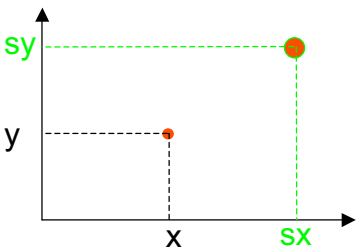
- Rotation:

$$p' = \begin{pmatrix} \cos\alpha \cdot x - \sin\alpha \cdot y \\ \sin\alpha \cdot x + \cos\alpha \cdot y \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{R}_\alpha \cdot p$$



- Skalierung:

$$p' = s \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix}$$



Transformationen im 3D-Raum

- Punktdarstellung: $p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

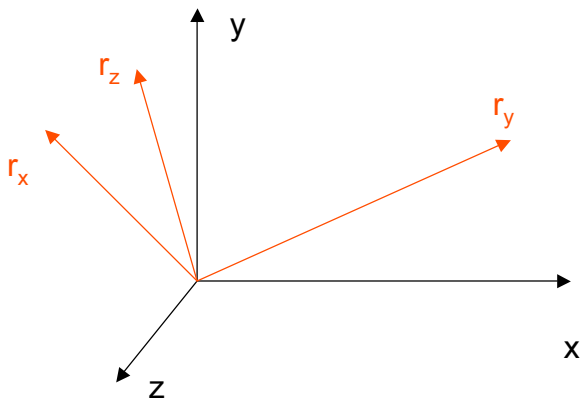
- Translation: $p' = p + t = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \end{pmatrix}$

- Skalierung: $p' = s \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} sx \\ sy \\ sz \end{pmatrix}$

Rotationen im 3D-Raum (1)

- Matrixdarstellung:

$$p' = \mathbf{R} \cdot p = \begin{pmatrix} r_x^T \\ r_y^T \\ r_z^T \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_x^1 & r_x^2 & r_x^3 \\ r_y^1 & r_y^2 & r_y^3 \\ r_z^1 & r_z^2 & r_z^3 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_x^1 x + r_x^2 y + r_x^3 z \\ r_y^1 x + r_y^2 y + r_y^3 z \\ r_z^1 x + r_z^2 y + r_z^3 z \end{pmatrix}$$



Rotationen im 3D-Raum (2)

- Eigenschaften von Rotationsmatrizen:
  - Orthonormal:
    $r_{i,j} \cdot r_{i,k} = \delta_{j,k}$
  - Also auch:
    $\mathbf{R}^{-1} = \mathbf{R}^{T}; \qquad \mathbf{R}^{T}\mathbf{R} = \mathbf{I}$
  - Determinante +1, sonst (-1) *Rotoinversion*, d.h. Rotation gefolgt von Spiegelung (führt rechts- in linkshändiges System über)
- Alle orthonormalen 3x3 Matrizen mit Determinante +1 bilden die Gruppe SO(3)

Rotationen im 3D-Raum (3)

- Vorteile der Matrixdarstellung:
  - Recht intuitiv
  - Einfache, lineare Berechnungsvorschrift
  - Eindeutige Darstellung
- Probleme der Matrixdarstellung:
  - Matrix muss orthonormal sein (problematisch bei numerischen Instabilitäten) → hohe Redundanz in Darstellung, 9 Parameter für 3 Freiheitsgrade
  - Keine einfache Interpolation von Rotationen

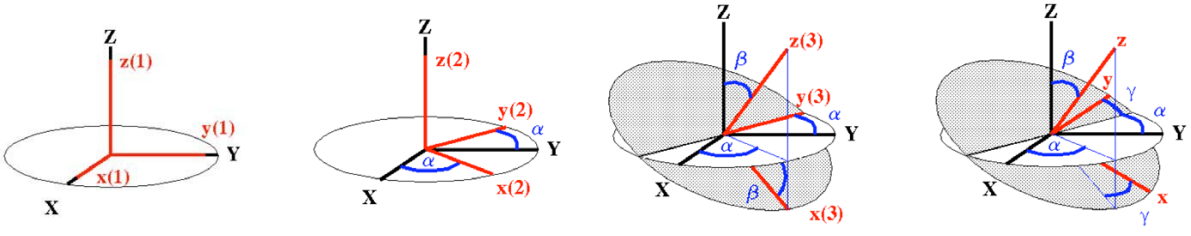
Rotationen im 3D-Raum (4)

- Rotationsmatrizen:

$$\mathbf{R}_\alpha, \mathbf{R}_\beta, \mathbf{R}_\gamma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}, \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}, \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotationen im 3D-Raum (5)

- Eulerwinkel:
  - Jede Rotation kann als Folge dreier Rotationen um drei Koordinatenachsen ausgedrückt werden
  - Häufigste Darstellung:
    α um z-Achse, dann β um *neue* y-Achse, dann γ um *ganz neue* z-Achse

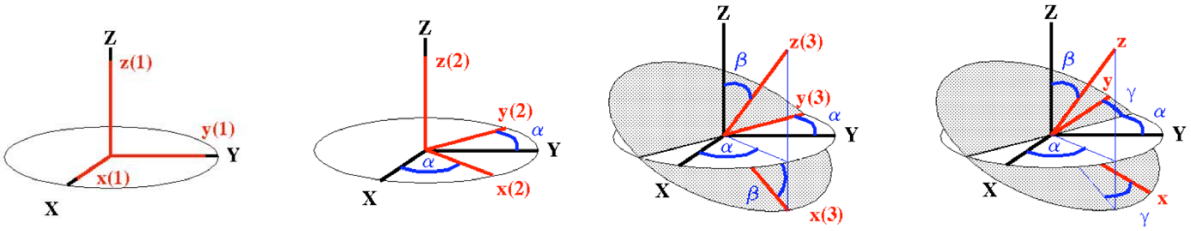


Rotationen im 3D-Raum (6)

- Eulerwinkel in Matrixdarstellung:

$$\mathbf{R} = \mathbf{R}_\alpha \cdot \mathbf{R}_\beta \cdot \mathbf{R}_\gamma = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \cdot \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\beta\cos\gamma \\ -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\beta\sin\gamma \\ \cos\alpha\sin\beta & \sin\alpha\sin\beta & \cos\beta \end{pmatrix}$$



Rotationen im 3D-Raum (7)

- Einziger Vorteil der Eulerwinkel:
  - Häufig verwendet, v.a. in traditionellen Trackinganwendungen der Luftfahrt („Roll/Pitch/Yaw“ – Rollen/Neigen/Gieren)
- Nachteile der Eulerwinkel:
  - Darstellung alles andere als eindeutig (Welche Achsenfolge? Neue/alte Achsen als Bezug?)
  - Nur auf den ersten Blick intuitiv
  - Möglicher Verlust eines Freiheitsgrades, wenn nach einer Rotation eine „neue“ auf eine „alte“ Achse fällt (*Gimbal lock*, von „Gimbal“ = Kardanring bei Gyroskopen)
  - Teure Berechnung: Sinus/Kosinus

Rotationen im 3D-Raum (8)

- Axis/Angle Darstellung:
  - Jede Rotation kann als Drehung um eine *bestimmte Achse x* und einen *bestimmten Winkel φ* dargestellt werden (Eulers Theorem)
  - Da $\boldsymbol{R}x = x$, ist x Eigenvektor zum Eigenwert 1 der äquivalenten Rotationsmatrix $\boldsymbol{R}$ (1 ist einziger realer Eigenwert von $\boldsymbol{R}$).
- Vorteile:
  - Intuitiv, weitgehend eindeutig
  - Geringe Redundanz (4 Parameter, 3 Freiheitsgrade, mit Normierung von x reichen 3 Parameter)
- Problem:
  - Kombination mehrerer Rotationen schwierig

Rotationen im 3D-Raum (9)

- Quaternionen (Hamilton, 1843):
  - Hyperkomplexe Zahlen vom Rang 4, bestehend aus Skalar q_0 und Vektor $\mathbf{q}$
  - Mit Einheitsquaternionen können Rotationen im 3D ausgedrückt werden
  - Berechung aus Axis/Angle $\mathbf{x}/\varphi$:

$$q_0 = \cos\frac{\varphi}{2}; \qquad \mathbf{q} = \sin\frac{\varphi}{2}\cdot\mathbf{x}$$

  - Vor-/Nachteile:
    - wie Axis/Angle, aber jetzt schnelle Kombination von Rotationen durch *Quaternionenmultiplikation*
    - Interpolation leicht möglich (→ Übung)

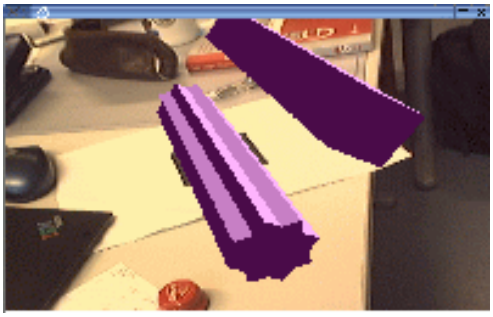
Rotationen in 3D: Zusammenfassung

- Verschiedene Darstellungsmöglichkeiten
- Orthonormale Rotationsmatrizen und Quaternionen vorteilhaft
  - Eindeutige Darstellung
  - Leichte/effiziente Kombination mehrerer Rotationen
- Vorsicht bei Eulerwinkeln!

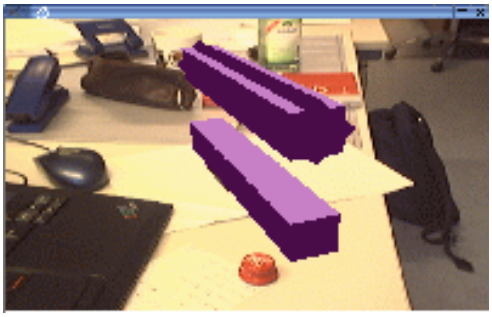
Kombination von Transformationen

- Eigentlich kein Problem, aber…
  die Reihenfolge macht den Unterschied:

Zuerst Rotation, dann Translation:



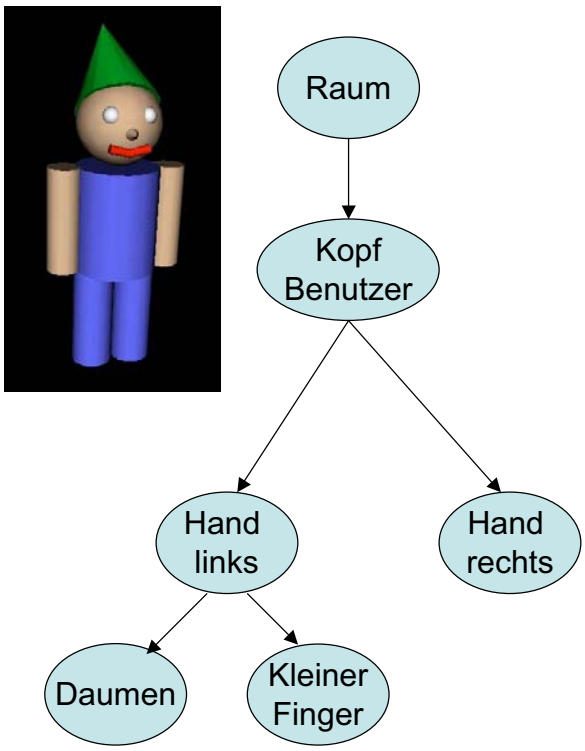
Zuerst Translation, dann Rotation:



- Inversion von Transformationen: Wechsel des Bezugssystems

Szenengraph

- Hierarchische Kombination von Transformationen (Konzept aus Computergraphik)
- Beispiel: Benutzer ist in einem Raum, Kamera auf seinem Kopf wird getrackt. Aus dem Kamerabild wird zudem die Position der Hände des Benutzers bestimmt.
- Hauptvorteil: Gruppierung von Objekten



Homogene Transformationsmatrizen (1)

- Wie kann man Kombination aus Translationen, Rotationen und Skalierungen als einzige Transformation darstellen?
- Rotationen und Skalierungen:
  - Darstellung als Matrix
  - Kombination: Produkt mehrerer Matrizen
- Translation:
  - Darstellung als Vektor
  - Kombination mehrerer Translationen: Summe der Vektoren
  - Kombination mit Rotations-/Skalierungsmatrix ??

Homogene Transformationsmatrizen (2)

- Durch 4. Komponente („homogene Vektoren“) können auch Translationen als Matrix dargestellt werden:

$$p = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}; \qquad \mathbf{H}_t = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad \mathbf{H}_t \cdot p = \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{pmatrix}$$

- Kombination mit Rotationen/Skalierungen durch einfaches Aufmultiplizieren:

$$\mathbf{H}_r = \begin{pmatrix} r_x^1 & r_y^1 & r_z^1 & 0 \\ r_x^2 & r_y^2 & r_z^2 & 0 \\ r_x^3 & r_y^3 & r_z^3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad \mathbf{H}_s = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Homogene Transformationsmatrizen (3)

- Homogene Matrizen sind Standard in der Computergraphik (z.B. OpenGL)
- Weitere Berechnungen möglich:
  - Projektionen in 2D (perspektivisch/orthogonal)
  - Ebenen in 3D und Schnitte

Räumliche Beziehungen

- Zurück zum Tracking:
  Bestimmung räumlicher Beziehungen
- Klassifikation:
  - Absolute vs. relative Ortsbestimmung
  - Ableitungen: Geschwindigkeit, Beschleunigung
- Am wichtigsten & häufigsten:
  Absolute Position und Orientierung im 3D-Raum

Anforderungen an AR-Tracker

- Hohe Genauigkeit
- Geringe Latenzzeit
- Hohe Wiederholrate (min. 10 fps)
- Kleine Baugröße, v.a. von mobilen Bestandteilen
- Meist 6 DOF, absolute Messung
- Simultane Unterstützung mehrerer Objekte/Benutzer
- Niedriger Preis

→ Gesucht: eierlegende Tracking-Wollmilchsau.

Mögliche Fehlerquellen (1)

- Latenz ist DIE Fehlerquelle
  - Jede Millisekunde führt max. zu 1mm, im Schnitt zu 1/3mm Registrierungsfehler
  - D.h.: ein System, das eine Genauigkeit von 1mm haben soll, darf eine *Gesamtlatenzzeit* von 1ms nicht überschreiten
- Ursachen der Latenz
  - Tracker
  - Bearbeitungszeit auf dem Host
  - Bildgenerierung
  - Diverse Displayeigenschaften

Mögliche Fehlerquellen (2)

- Numerische Fehler bei der Akkumulation von Koordinatensystemen
- Optische Fehler: Kissenverzerrung, fehlendes Augentracking
- Trackermessfehler:
  - Statische Fehler (unveränderlich, können durch gute Kalibrierung behoben werden)
  - Jitter (Rauschen, kann nicht behoben werden)
  - Dynamische Fehler (abhängig von der Bewegung des Sensors und/oder getrackten Objekts)